

# **Doxxygen\_MPDASTARONLINE\_DB Reference Manual**

Generated by Doxygen 1.3.7

Wed Sep 6 11:55:13 2006



# Contents

<b>1</b>	<b>Doxxygen_MPDAUTHORITYDB Class Index</b>	<b>1</b>
1.1	Doxygen_MPDAUTHORITYDB Class List . . . . .	1
<b>2</b>	<b>Doxxygen_MPDAUTHORITYDB File Index</b>	<b>3</b>
2.1	Doxygen_MPDAUTHORITYDB File List . . . . .	3
<b>3</b>	<b>Doxxygen_MPDAUTHORITYDB Class Documentation</b>	<b>5</b>
3.1	rhicBeamSender Class Reference . . . . .	5
3.2	rhicScalarsSender Class Reference . . . . .	16
3.3	starMagnetSender Class Reference . . . . .	27
<b>4</b>	<b>Doxxygen_MPDAUTHORITYDB File Documentation</b>	<b>39</b>
4.1	rhicBeamDaemon.cc File Reference . . . . .	39
4.2	rhicBeamSender.cc File Reference . . . . .	40
4.3	rhicBeamSender.hh File Reference . . . . .	41
4.4	rhicBeamSender_i.cc File Reference . . . . .	42
4.5	rhicScalarsDaemon.cc File Reference . . . . .	43
4.6	rhicScalarsSender.cc File Reference . . . . .	44
4.7	rhicScalarsSender.hh File Reference . . . . .	45
4.8	rhicScalarsSender_i.cc File Reference . . . . .	46
4.9	sms_i.cc File Reference . . . . .	47
4.10	starMagnetDaemon.cc File Reference . . . . .	48
4.11	starMagnetSender.cc File Reference . . . . .	49
4.12	starMagnetSender.hh File Reference . . . . .	50
4.13	starMagnetSender_i.cc File Reference . . . . .	51



# Chapter 1

## **Doxxygen\_MPDASTARONLINE\_DB Class Index**

### **1.1 Doxygen\_MPDASTARONLINE\_DB Class List**

Here are the classes, structs, unions and interfaces with brief descriptions:

rhicBeamSender	5
rhicScalarsSender	16
starMagnetSender	27



## Chapter 2

# Doxxygen\_MPDAEMON\_StarOnlineDB File Index

### 2.1 Doxygen\_MPDAEMON\_StarOnlineDB File List

Here is a list of all files with brief descriptions:

rhicBeamDaemon.cc . . . . .	39
rhicBeamSender.cc . . . . .	40
rhicBeamSender.hh . . . . .	41
rhicBeamSender_i.cc . . . . .	42
rhicScalarsDaemon.cc . . . . .	43
rhicScalarsSender.cc . . . . .	44
rhicScalarsSender.hh . . . . .	45
rhicScalarsSender_i.cc . . . . .	46
sms_i.cc . . . . .	47
starMagnetDaemon.cc . . . . .	48
starMagnetSender.cc . . . . .	49
starMagnetSender.hh . . . . .	50
starMagnetSender_i.cc . . . . .	51



# Chapter 3

## Doxxygen\_MP\_D\_STAR\_ONLINE\_DB Class Documentation

### 3.1 rhicBeamSender Class Reference

```
#include <rhicBeamSender.hh>
```

#### Public Member Functions

- `rhicBeamSender` (const char \*localDir)  
*dito*
  - `virtual ~rhicBeamSender ()`
  - `virtual void initTable ()`
  - `virtual void initTags ()`
  - `virtual void initDataBase ()`
  - `virtual bool loadUserControls (const char *name, const char *value)`
  - `virtual void initQuery ()`
  - `virtual bool queryData ()`
  - `virtual bool readData (const char *fileName)`
  - `virtual bool updateDb (const char *fileName)`
  - `virtual bool readData (ifstream &from)`
  - `virtual bool hasChanged (int rowNumber)`
  - `char * readAny ()`
  - `bool readVal (char *&value)`
  - `bool readVal (float &value)`
  - `bool readVal (double &value)`
  - `bool readVal (short &value)`
  - `bool readVal (int &value)`
  - `bool readVal (long &value)`
  - `bool readVal (long long &value)`
  - `bool nextLine (ifstream &from)`
  - `void readError (int l, char *c, char *m)`

## Protected Attributes

- rhicBeam `previousVals` [NUM\_DB\_ROWS]
- rhicBeam `tempVals` [NUM\_DB\_ROWS]
- int `elementList` [NUM\_DB\_ROWS]
- rhicBeam `updateVals` [NUM\_DB\_ROWS]
- int `updateElements` [NUM\_DB\_ROWS]
- bool `mreadStatus`
- char `mline` [256]
- char `tmpline` [256]
- char \* `ptr1`
- char \* `ptr2`

### 3.1.1 Constructor & Destructor Documentation

#### 3.1.1.1 `rhicBeamSender::rhicBeamSender (const char * localDir)`

dito

Definition at line 18 of file rhicBeamSender.cc.

```

18
19
20     initTags();
21     if(localDir) cd(localDir); // note this ignores the sub dir tag
22     init("rhicBeam"); // setup the file I/O
23     initDataBase(); // database connections
24     initTable(); // table definitions
25
26 }
```

#### 3.1.1.2 `virtual rhicBeamSender::~rhicBeamSender () [inline, virtual]`

Definition at line 41 of file rhicBeamSender.hh.

```
41 {};
```

### 3.1.2 Member Function Documentation

#### 3.1.2.1 `bool rhicBeamSender::hasChanged (int rowNumber) [virtual]`

Definition at line 153 of file rhicBeamSender\_i.cc.

```

153
154
155 rhicBeam* pre=&previousVals[rowNumber];
156 rhicBeam* cur=&tempVals[rowNumber];
157
158 /* example ... note -> change to any element requires db-update
159 * and thus returns true immediately
160 *
161 *if(fabs(pre->ch0Voltage-cur->ch0Voltage)>=driftLimit) return true;
162 *if(fabs(pre->ch1Voltage-cur->ch1Voltage)>=driftLimit) return true;
163 *
```

```

164 *    ....
165 */
166
167 return true;
168 }
```

### 3.1.2.2 void rhicBeamSender::initDataBase () [virtual]

Definition at line 74 of file rhicBeamSender.cc.

```

74 {
75 #define __METHOD__ "initDataBase()"
76
77 /* More than an example... swap user & dbTrg as per subsystem*/
78 mgr->setUser("stardb","");
79 StDbType dbT = dbConditions;
80 StDbDomain dbD = dbRhic;
81
82 if( !( node = mgr->initConfig(dbT,dbD) ) )
83     sendMess("Connect Failed ",mgr->printDbName(dbT,dbD),dbMFatal,__LINE__,__CLASS__,__METHOD__);
84
85 #undef __METHOD__
86 }
```

### 3.1.2.3 void rhicBeamSender::initQuery () [virtual]

Definition at line 38 of file rhicBeamSender\_i.cc.

```

38 {
39 #define __METHOD__ "initQuery()"
40
41 ofstream to(queryFile);
42
43 if(!to.is_open()){
44     sendMess("Open Failed ",queryFile,dbMFatal,__LINE__,__CLASS__,__METHOD__);
45     return;
46 }
47
48 to<<"cdev_bluebeam_particle"<<endl;
49 to<<"cdev_bluebeam_energy"<<endl;
50 to<<"cdev_bluebeam_ions"<<endl;
51 to<<"cdev_bluebeam_lifetime"<<endl;
52 to<<"cdev_bluebeam_wcm"<<endl;
53 to<<"cdev_yellowbeam_particle"<<endl;
54 to<<"cdev_yellowbeam_energy"<<endl;
55 to<<"cdev_yellowbeam_ions"<<endl;
56 to<<"cdev_yellowbeam_lifetime"<<endl;
57 to<<"cdev_yellowbeam_wcm"<<endl;
58 to<<"cdev_blue-yellow_g5bx_H"<<endl;
59 to<<"cdev_blue-yellow_g5bx_V"<<endl;
60 to<<"cdev_blue-yellow_g6bx_H"<<endl;
61 to<<"cdev_blue-yellow_g6bx_V"<<endl;
62 to<<"cdev_bluebeam_fillNumber"<<endl;
63 to<<"cdev_yellowbeam_fillNumber"<<endl;
64
65
66 /* example
67 *      for(int i=0;i<16;i++){
68 *          to<<"TRGhv:SUB_RD_V_1:<<i<<.E"<<endl;
69 *          to<<"TRGhv:SUB_RD_V_1:<<i<<.F"<<endl;
70 *          ....
```

```

71  *
72  *      }
73  */
74
75     to.close();
76
77 #undef __METHOD__
78 }
```

### 3.1.2.4 void rhicBeamSender::initTable () [virtual]

Definition at line 29 of file rhicBeamSender.cc.

```

29
30 #define __METHOD__ "initTable()"
31
32     StDbTable* table=0;
33     if(! (table=node->addDbTable("rhicBeam")))
34         sendMess("Could not find table=rhicBeam",dbMFatal,__LINE__,__CLASS__,__METHOD__);
35
36     memset(tempVals,0,NUM_DB_ROWS*sizeof(rhicBeam));
37     memset(previousVals,0,NUM_DB_ROWS*sizeof(rhicBeam));
38
39     int nrows;
40     int* elist = table->getElementID(nrows);
41     if(nrows!=NUM_DB_ROWS){
42         //char mess[256];
43         ostringstream ms;
44         ms<<"Db rows("<<nrows<<") != compiled(" <<ends;
45         sendMess((ms.str()).c_str(),dbMFatal,__LINE__,__CLASS__,__METHOD__);
46     }
47     memcpy(elementList,elist,NUM_DB_ROWS*sizeof(int));
48
49     unsigned int timestamp=time(NULL);
50     mgr->setRequestTime(timestamp);
51     if(mgr->fetchDbTable(table)){
52         rhicBeam* thv = (rhicBeam*)table->GetTable();
53         memcpy(previousVals,thv,nrows*sizeof(rhicBeam));
54     }
55
56 #undef __METHOD__
57 };
```

### 3.1.2.5 void rhicBeamSender::initTags () [virtual]

Definition at line 65 of file rhicBeamSender.cc.

```

65
66 /* more than an example -> swap "trg" to your subsys & add to email list*/
67     setEmailTo("porter@bnl.gov");
68     setDomainName("rhic");
69
70 }
```

### 3.1.2.6 bool rhicBeamSender::loadUserControls (const char \* name, const char \* value) [virtual]

Definition at line 20 of file rhicBeamSender\_i.cc.

```

20                                         {
21 #define __METHOD__ "loadUserControls(name,value)"
22
23 /* more than an example ... swap driftLimit to yours
24 * and duplicate this structure for each selection criteria
25 if(strstr(name,"driftLimit")){
26     driftLimit=atof(value);
27     sendMess("driftLimit set=",value,dbMDebug,__LINE__,__CLASS__,__METHOD__);
28     return true;
29 }
30 */
31
32 return false;
33 #undef __METHOD__
34 }
```

### 3.1.2.7 bool rhicBeamSender::nextLine (ifstream &*from*) [inline]

Definition at line 74 of file rhicBeamSender.hh.

```

74                                         {
75     if(!from.getline(mline,255))return false;
76     return true;
77 }
```

### 3.1.2.8 bool rhicBeamSender::queryData () [virtual]

Definition at line 90 of file rhicBeamSender.cc.

```

90                                         {
91 #define __METHOD__ "queryData()"
92
93 /*
94  * MORE THAN AN EXAMPLE....
95  * IF Standard SC-Query via "caGet" then,
96  * no need to change this method AT ALL
97  *
98 */
99
100    writeTime = (unsigned int)time(NULL);           //for database write time
101
102    //char systemCmd[1024];
103    ostringstream scmd;
104    scmd<<"caGet "<<queryFile<<" "<<dataFile<<ends;
105
106    if(system((scmd.str()).c_str()))
107        return sendMess(" caGet system call returned error",dbMErr,__LINE__,__CLASS__,__METHOD__);
108
109    return true;
110 #undef __METHOD__
111 };
```

### 3.1.2.9 char \* rhicBeamSender::readAny ()

Definition at line 196 of file rhicBeamSender.cc.

```

196
197
198     strcpy(tmpLine,mLine);
199     pTr1=tmpLine;
200     pTr2=strtok(pTr1," ");
201     if(!pTr2) return pTr2;
202     pTr2=strtok(NULL," ");
203     return pTr2;
204 }
```

### 3.1.2.10 bool rhicBeamSender::readData (ifstream & from) [virtual]

Definition at line 83 of file rhicBeamSender\_i.cc.

```

83
84 #define __METHOD__ "readData(ifstream)"
85
86 mReadStatus=true;
87 memset(tempVals,0,NUM_DB_ROWS*sizeof(rhicBeam));
88
89
90 int i=0;
91 char* c=__CLASS__;
92 char* m=__METHOD__;
93
94 //MPD Changes
95     char* tmpVal=new char[32];
96
97     memset(tmpVal,0,32*sizeof(char));
98     if(!nextLine(from) || !readVal(tmpVal))readError(__LINE__,c,m);
99     if(tmpVal[0]!='\0')strcpy(tempVals[i].blueSpecies,tmpVal);
100 //****old change --delete when works
101     if(!nextLine(from)){
102         tmpVal0=tempVals[i].blueSpecies;
103         if(!readVal(tmpVal0))readError(__LINE__,c,m);
104     }/*
105 // if(!nextLine(from) || !readVal((char*)tempVals[i].blueSpecies)) readError(__LINE__,c,m);
106 if(!nextLine(from) || !readVal(tempVals[i].blueEnergy)) readError(__LINE__,c,m);
107 if(!nextLine(from) || !readVal(tempVals[i].blueIons)) readError(__LINE__,c,m);
108 if(!nextLine(from) || !readVal(tempVals[i].blueLifeTime)) readError(__LINE__,c,m);
109 if(!nextLine(from) || !readVal(tempVals[i].blueBunchIons)) readError(__LINE__,c,m);
110 //MPD Changes
111     char* tmpVal1=new char[32];
112
113     memset(tmpVal1,0,32*sizeof(char));
114     if(!nextLine(from) || !readVal(tmpVal1))readError(__LINE__,c,m);
115     if(tmpVal1[0]!='\0')strcpy(tempVals[i].yellowSpecies,tmpVal1);
116
117 //****old changes -- delete when works
118     if(!nextLine(from)){
119         tmpVal1=tempVals[i].yellowSpecies;
120         if(!readVal(tmpVal1))readError(__LINE__,c,m);
121     }/*
122 //if(!nextLine(from) || !readVal((char*)tempVals[i].yellowSpecies)) readError(__LINE__,c,m);
123 if(!nextLine(from) || !readVal(tempVals[i].yellowEnergy)) readError(__LINE__,c,m);
124 if(!nextLine(from) || !readVal(tempVals[i].yellowIons)) readError(__LINE__,c,m);
125 if(!nextLine(from) || !readVal(tempVals[i].yellowLifeTime)) readError(__LINE__,c,m);
126 if(!nextLine(from) || !readVal(tempVals[i].yellowBunchIons)) readError(__LINE__,c,m);
127 if(!nextLine(from) || !readVal(tempVals[i].byHorzDiffAt50C)) readError(__LINE__,c,m);
128 if(!nextLine(from) || !readVal(tempVals[i].byVertDiffAt50C)) readError(__LINE__,c,m);
129 if(!nextLine(from) || !readVal(tempVals[i].byHorzDiffAt60C)) readError(__LINE__,c,m);
130 if(!nextLine(from) || !readVal(tempVals[i].byVertDiffAt60C)) readError(__LINE__,c,m);
131 if(!nextLine(from) || !readVal(tempVals[i].blueFillNumber)) readError(__LINE__,c,m);
132 if(!nextLine(from) || !readVal(tempVals[i].yellowFillNumber)) readError(__LINE__,c,m);
```

```

133
134
135 /* example format
136 *   for(int i=0;i<NUM_DB_ROWS;i++){
137 *     if(!nextLine(from) || !readVal(tempVals[i].blah)) readError(l,c,m);
138 *
139 *     ....
140 *
141 *   }
142 */
143
144   from.close();
145   delete tmpval;
146   delete tmpval1;
147 return true;
148 #undef __METHOD__
149 }
```

**3.1.2.11 bool rhicBeamSender::readData (const char \*fileName) [virtual]**

Definition at line 115 of file rhicBeamSender.cc.

```

115
116 #define __METHOD__ "readData(fileName)"
117
118   ifstream from(fileName);
119   if(!from) return sendMess("Cannot open file=",fileName,dbMErr,__LINE__,__CLASS__,__METHOD__);
120
121   return readData(from); // user implemented file read
122 #undef __METHOD__
123 }
```

**3.1.2.12 void rhicBeamSender::readError (int l, char \* c, char \* m) [inline]**

Definition at line 79 of file rhicBeamSender.hh.

```

79
80   mreadStatus=sendMess(" *** Missing Data at ",mline,dbMErr,l,c,m);
81 }
```

**3.1.2.13 bool rhicBeamSender::readVal (long long & value)**

Definition at line 268 of file rhicBeamSender.cc.

```

268
269
270   if(!readAny())return false;
271   char* store[256];
272   value=strtoll(ptr2,store,10);
273   if(strlen(*store)>0) return false; // value is not a number
274
275   return true;
276 };
```

### 3.1.2.14 bool rhicBeamSender::readVal (long & *value*)

Definition at line 257 of file rhicBeamSender.cc.

```
257
258
259     if(!readAny())return false;
260
261     char* store[256];
262     value=strtol(ptr2,store,10);
263     if(strlen(*store)>0) return false; // value is not a number
264
265     return true;
266 };
```

### 3.1.2.15 bool rhicBeamSender::readVal (int & *value*)

Definition at line 246 of file rhicBeamSender.cc.

```
246
247
248     if(!readAny()) return false;
249
250     char* store[256];
251     value=(int)strtol(ptr2,store,10);
252     if(strlen(*store)>0) return false; // value is not a number
253
254     return true;
255 };
```

### 3.1.2.16 bool rhicBeamSender::readVal (short & *value*)

Definition at line 235 of file rhicBeamSender.cc.

```
235
236
237     if(!readAny()) return false;
238
239     char* store[256];
240     value=(short)strtol(ptr2,store,10);
241     if(strlen(*store)>0) return false; // value is not a number
242
243     return true;
244 };
```

### 3.1.2.17 bool rhicBeamSender::readVal (double & *value*)

Definition at line 224 of file rhicBeamSender.cc.

```
224
225
226     if(!readAny())return false;
227
228     char* store[256];
229     value=strtod(ptr2,store);
230     if(strlen(*store)>0) return false; // value is not a number
231
232     return true;
233 };
```

**3.1.2.18 bool rhicBeamSender::readVal (float & *value*)**

Definition at line 213 of file rhicBeamSender.cc.

```

213
214
215     if(!readAny()) return false;
216
217     char* store[256];
218     value=(float)strtod(ptr2,store);
219     if(strlen(*store)>0) return false; // value is not a number
220
221     return true;
222 };

```

**3.1.2.19 bool rhicBeamSender::readVal (char \*& *value*)**

Definition at line 206 of file rhicBeamSender.cc.

```

206
207
208     if(!readAny()) return false;
209     strcpy(value,ptr2);
210     return true;
211 }

```

**3.1.2.20 bool rhicBeamSender::updateDb (const char \**fileName*) [virtual]**

Definition at line 126 of file rhicBeamSender.cc.

```

126
127 #define __METHOD__ "updateDb(filename)"
128
129     if(!readData(fileName)) return sendMess(" Read data failed",dbMErr,__LINE__,__CLASS__,__METHOD__);
130
131     int* elements;
132     rhicBeam* vals;
133     int numRows = 0;
134
135     if(writeRequired()){
136
137         numRows=NUM_DB_ROWS;
138         elements=elementList;
139         vals = tempVals;
140
141     } else {
142
143         for(int i=0; i<NUM_DB_ROWS; i++){
144             if(hasChanged(i)){
145                 updateElements[numRows]=elementList[i];
146                 updateVals[numRows] = tempVals[i];
147                 previousVals[i]=tempVals[i];
148                 numRows++;
149             }
150         }
151
152         elements = updateElements;
153         vals      = updateVals;
154     }

```

```

155
156     if(numRows==0) return sendMess(" No update required for ",mbaseName,dbMDebug,__LINE__,__CLASS__,__METHOD__);
157
158     //char mess[256];
159     ostringstream sn;
160     sn<<"Will Update "<<numRows<<" of "<<NUM_DB_ROWS<<" rows "<<ends;
161     sendMess((sn.str()).c_str(),dbMDebug,__LINE__,__CLASS__,__METHOD__);
162
163     StDbTable* dbTable=node->findTable("rhicBeam");
164     dbTable->SetTable((char*)vals, numRows, elements);
165     mgr->setStoreTime(writeTime);
166
167     if(!mgr->storeDbTable(dbTable)) {
168         addBackLog(writeTime);
169         return sendMess("Store failed ",dbMErr,__LINE__,__CLASS__,__METHOD__);
170     }
171
172     if(numRows==NUM_DB_ROWS)lastFullWrite=writeTime;
173
174     return true;
175 #undef __METHOD__
176 }
```

### 3.1.3 Member Data Documentation

#### 3.1.3.1 int rhicBeamSender::elementList[NUM\_DB\_ROWS] [protected]

Definition at line 24 of file rhicBeamSender.hh.

#### 3.1.3.2 char rhicBeamSender::mline[256] [protected]

Definition at line 29 of file rhicBeamSender.hh.

#### 3.1.3.3 bool rhicBeamSender::mreadStatus [protected]

Definition at line 28 of file rhicBeamSender.hh.

#### 3.1.3.4 rhicBeam rhicBeamSender::previousVals[NUM\_DB\_ROWS] [protected]

Definition at line 22 of file rhicBeamSender.hh.

#### 3.1.3.5 char\* rhicBeamSender::ptr1 [protected]

Definition at line 31 of file rhicBeamSender.hh.

#### 3.1.3.6 char \* rhicBeamSender::ptr2 [protected]

Definition at line 31 of file rhicBeamSender.hh.

#### 3.1.3.7 rhicBeam rhicBeamSender::tempVals[NUM\_DB\_ROWS] [protected]

Definition at line 23 of file rhicBeamSender.hh.

**3.1.3.8 char rhicBeamSender::tmpline[256] [protected]**

Definition at line 30 of file rhicBeamSender.hh.

**3.1.3.9 int rhicBeamSender::updateElements[NUM\_DB\_ROWS] [protected]**

Definition at line 26 of file rhicBeamSender.hh.

**3.1.3.10 rhicBeam rhicBeamSender::updateVals[NUM\_DB\_ROWS] [protected]**

Definition at line 25 of file rhicBeamSender.hh.

The documentation for this class was generated from the following files:

- [rhicBeamSender.hh](#)
- [rhicBeamSender.cc](#)
- [rhicBeamSender\\_i.cc](#)

## 3.2 rhicScalarsSender Class Reference

```
#include <rhicScalarsSender.hh>
```

### Public Member Functions

- `rhicScalarsSender` (const char \*localDir)
- virtual ~`rhicScalarsSender` ()
- virtual void `initTable` ()
- virtual void `initTags` ()
- virtual void `initDataBase` ()
- virtual bool `loadUserControls` (const char \*name, const char \*value)
- virtual void `initQuery` ()
- virtual bool `queryData` ()
- virtual bool `readData` (const char \*fileName)
- virtual bool `updateDb` (const char \*fileName)
- virtual bool `readData` (ifstream &from)
- virtual bool `hasChanged` (int rowNumber)
- char \* `readAny` ()
- bool `readVal` (char \*&value)
- bool `readVal` (float &value)
- bool `readVal` (double &value)
- bool `readVal` (short &value)
- bool `readVal` (int &value)
- bool `readVal` (unsigned int &value)
- bool `readVal` (long &value)
- bool `readVal` (long long &value)
- bool `nextLine` (ifstream &from)
- void `readError` (int l, char \*c, char \*m)

### Protected Attributes

- rhicScalars `previousVals` [NUM\_DB\_ROWS]
  - rhicScalars `tempVals` [NUM\_DB\_ROWS]
  - int `elementList` [NUM\_DB\_ROWS]
  - rhicScalars `updateVals` [NUM\_DB\_ROWS]
  - int `updateElements` [NUM\_DB\_ROWS]
  - bool `mreadStatus`
  - char `mline` [256]
  - char `tmpline` [256]
  - char \* `ptr1`
  - char \* `ptr2`
  - int `countLimit`
- dito*
- unsigned int `lastTime`

### 3.2.1 Constructor & Destructor Documentation

#### 3.2.1.1 rhicScalarsSender::rhicScalarsSender (const char \* *localDir*)

Definition at line 18 of file rhicScalarsSender.cc.

```

18
19
20     initTags();
21     if(localDir) cd(localDir); // note this ignores the sub dir tag
22     init("rhicScalars"); // setup the file I/O
23     initDataBase();        // database connections
24     initTable();           // table definitions
25
26 }
```

#### 3.2.1.2 virtual rhicScalarsSender::~rhicScalarsSender () [inline, virtual]

Definition at line 43 of file rhicScalarsSender.hh.

```
43 {};
```

### 3.2.2 Member Function Documentation

#### 3.2.2.1 bool rhicScalarsSender::hasChanged (int *rowNumber*) [virtual]

Definition at line 117 of file rhicScalarsSender\_i.cc.

```

117
118
119 rhicScalars* pre=&previousVals[rowNumber];
120 rhicScalars* cur=&tempVals[rowNumber];
121
122 if( (cur->scalar02-pre->scalar02) < 0 ||
123     (cur->scalar01-pre->scalar01) < 0){
124
125     cur->dscalar01=cur->scalar01;
126     cur->dscalar02=cur->scalar02;
127     cur->dscalar03=cur->scalar03;
128     cur->dscalar04=cur->scalar04;
129     cur->dscalar05=cur->scalar05;
130     cur->dscalar06=cur->scalar06;
131     cur->dscalar07=cur->scalar07;
132     cur->dscalar08=cur->scalar08;
133     cur->dscalar09=cur->scalar09;
134     cur->dscalar10=cur->scalar10;
135     cur->dscalar11=cur->scalar11;
136     cur->dscalar12=cur->scalar12;
137     cur->dscalar13=cur->scalar13;
138     cur->dscalar14=cur->scalar14;
139     cur->dscalar15=cur->scalar15;
140     cur->dscalar16=cur->scalar16;
141
142     cur->timeInterval=writeTime-lastTime;
143     lastTime=writeTime;
144     return true;
145 }
146
147 // check the zdc singles
```

```

148 /*****removed 12/16
149 if( (abs(pre->scalar02-cur->scalar02)>=countLimit) ||
150   (abs(pre->scalar03-cur->scalar03)>=countLimit) ||
151   (abs(pre->scalar12-cur->scalar12)>=countLimit) ||
152   (abs(pre->scalar13-cur->scalar13)>=countLimit) ||
153   (abs(pre->scalar14-cur->scalar14)>=countLimit) ){
154 *****/
155   cur->dscalar01=cur->scalar01-pre->scalar01;
156   cur->dscalar02=cur->scalar02-pre->scalar02;
157   cur->dscalar03=cur->scalar03-pre->scalar03;
158   cur->dscalar04=cur->scalar04-pre->scalar04;
159   cur->dscalar05=cur->scalar05-pre->scalar05;
160   cur->dscalar06=cur->scalar06-pre->scalar06;
161   cur->dscalar07=cur->scalar07-pre->scalar07;
162   cur->dscalar08=cur->scalar08-pre->scalar08;
163   cur->dscalar09=cur->scalar09-pre->scalar09;
164   cur->dscalar10=cur->scalar10-pre->scalar10;
165   cur->dscalar11=cur->scalar11-pre->scalar11;
166   cur->dscalar12=cur->scalar12-pre->scalar12;
167   cur->dscalar13=cur->scalar13-pre->scalar13;
168   cur->dscalar14=cur->scalar14-pre->scalar14;
169   cur->dscalar15=cur->scalar15-pre->scalar15;
170   cur->dscalar16=cur->scalar16-pre->scalar16;
171
172   cur->timeInterval=writeTime-lastTime;
173   lastTime=writeTime;
174   return true;
175 }
176
177 /* example ... note -> change to any element requires db-update
178 * and thus returns true immediately
179 */
180 *if(fabs(pre->ch0Voltage-cur->ch0Voltage)>=driftLimit) return true;
181 *if(fabs(pre->ch1Voltage-cur->ch1Voltage)>=driftLimit) return true;
182 *
183 * ...
184 */
185
186 // return false;
187 }
```

### 3.2.2.2 void rhicScalarsSender::initDataBase () [virtual]

Definition at line 74 of file rhicScalarsSender.cc.

```

74 {
75 #define __METHOD__ "initDataBase()"
76
77 /* More than an example... swap user & dbTrg as per subsystem*/
78 mgr->setUser("stardb","");
79 StDbType dbT = dbConditions;
80 StDbDomain dbD = dbRhic;
81
82 if( !( node = mgr->initConfig(dbT,dbD)) )
83   sendMess("Connect Failed ",mgr->printDbName(dbT,dbD),dbMFatal,__LINE__,__CLASS__,__METHOD__);
84
85 #undef __METHOD__
86 }
```

### 3.2.2.3 void rhicScalarsSender::initQuery () [virtual]

Definition at line 43 of file rhicScalarsSender\_i.cc.

```

43
44 #define __METHOD__ "initQuery()"
45
46     ofstream to(queryFile);
47
48     if(!to.is_open()){
49         sendMess("Open Failed ",queryFile,dbMFatal,__LINE__,__CLASS__,__METHOD__);
50         return;
51     }
52
53     for(int i=1;i<17;i++)to<<"cdev_scalar"<<i<<endl;
54
55 /* example
56 *      for(int i=0;i<16;i++){
57 *          to<<"TRGhv:SUB_RD_V_1:"<<i<<".E"<<endl;
58 *          to<<"TRGhv:SUB_RD_V_1:"<<i<<".F"<<endl;
59 *          ....
60 *
61 *      }
62 */
63
64     to.close();
65
66 #undef __METHOD__
67 }
```

### 3.2.2.4 void rhicScalarsSender::initTable () [virtual]

Definition at line 29 of file rhicScalarsSender.cc.

```

29
30 #define __METHOD__ "initTable()"
31
32     StDbTable* table=0;
33     if(!(table=node->addDbTable("rhicScalars")))
34         sendMess("Could not find table=rhicScalars",dbMFatal,__LINE__,__CLASS__,__METHOD__);
35
36     memset(tempVals,0,NUM_DB_ROWS*sizeof(rhicScalars));
37     memset(previousVals,0,NUM_DB_ROWS*sizeof(rhicScalars));
38
39     int nrows;
40     int* elist = table->getElementID(nrows);
41     if(nrows!=NUM_DB_ROWS){
42         //char mess[256];
43         ostringstream ms;
44         ms<<"Db rows("<<nrows<<") != compiled("<<NUM_DB_ROWS<<") "<<ends;
45         sendMess((ms.str()).c_str(),dbMFatal,__LINE__,__CLASS__,__METHOD__);
46     }
47     memcpy(elementList,elist,NUM_DB_ROWS*sizeof(int));
48
49     unsigned int timestamp=time(NULL);
50     mgr->setRequestTime(timestamp);
51     if(mgr->fetchDbTable(table)){
52         rhicScalars* thv = (rhicScalars*)table->GetTable();
53         memcpy(previousVals,thv,nrows*sizeof(rhicScalars));
54     }
55
56 #undef __METHOD__
57 };
```

### 3.2.2.5 void rhicScalarsSender::initTags () [virtual]

Definition at line 65 of file rhicScalarsSender.cc.

```

65      {
66      /* more than an example -> swap "trg" to your subsys & add to email list*/
67      setEmailTo("porter@bnl.gov");
68      setDomainName("rhic");
69
70  }

```

### 3.2.2.6 **bool rhicScalarsSender::loadUserControls (const char \* *name*, const char \* *value*) [virtual]**

Definition at line 20 of file rhicScalarsSender\_i.cc.

```

20
21 #define __METHOD__ "loadUserControls(name,value)"           {
22
23 /* more than an example ... swap driftLimit to yours
24 * and duplicate this structure for each selection criteria
25 if strstr(name,"driftLimit")){
26     driftLimit=atof(value);
27     sendMess("driftLimit set=",value,dbMDebug,__LINE__,__CLASS__,__METHOD__);
28     return true;
29 }
30 */
31 if strstr(name,"countLimit")){
32     countLimit=atoi(value);
33     sendMess("countLimit set=",value,dbMDebug,__LINE__,__CLASS__,__METHOD__);
34     return true;
35 }
36
37 return false;
38 #undef __METHOD__
39 }

```

### 3.2.2.7 **bool rhicScalarsSender::nextLine (ifstream & *from*) [inline]**

Definition at line 77 of file rhicScalarsSender.hh.

```

77
78     if(!from.getline(mline,255))return false;
79     return true;
80 }

```

### 3.2.2.8 **bool rhicScalarsSender::queryData () [virtual]**

Definition at line 90 of file rhicScalarsSender.cc.

```

90
91 #define __METHOD__ "queryData()"
92
93 /*
94  * MORE THAN AN EXAMPLE....
95  * IF Standard SC-Query via "caGet" then,
96  * no need to change this method AT ALL
97  *
98 */
99
100    writeTime = (unsigned int)time(NULL);           //for database write time

```

```

101 //char systemCmd[1024];
102 ostringstream scmd;
103 scmd<<"caGet "<<queryFile<<" "<<dataFile<<ends;
105
106 if(system((scmd.str()).c_str()))
107     return sendMess(" caGet system call returned error",dbMErr,__LINE__,__CLASS__,__METHOD__);
108
109 return true;
110 #undef __METHOD__
111 };

```

### 3.2.2.9 char \* rhicScalarsSender::readAny ()

Definition at line 196 of file rhicScalarsSender.cc.

```

196 {
197
198     strcpy(tmpLine,mLine);
199     ptr1=tmpLine;
200     ptr2=strtok(ptr1," ");
201     if(!ptr2) return ptr2;
202     ptr2=strtok(NULL," ");
203     return ptr2;
204 }

```

### 3.2.2.10 bool rhicScalarsSender::readData (ifstream & from) [virtual]

Definition at line 72 of file rhicScalarsSender\_i.cc.

```

72                                         {
73 #define __METHOD__ "readData(ifstream)"
74
75 mreadStatus=true;
76 memset(tempVals,0,NUM_DB_ROWS*sizeof(rhicScalars));
77
78
79 int i=0;
80 char* c=__CLASS__;
81 char* m=__METHOD__;
82
83
84     if(!nextLine(from) || !readVal(tempVals[i].scalar01)) readError(__LINE__,c,m);
85     if(!nextLine(from) || !readVal(tempVals[i].scalar02)) readError(__LINE__,c,m);
86     if(!nextLine(from) || !readVal(tempVals[i].scalar03)) readError(__LINE__,c,m);
87     if(!nextLine(from) || !readVal(tempVals[i].scalar04)) readError(__LINE__,c,m);
88     if(!nextLine(from) || !readVal(tempVals[i].scalar05)) readError(__LINE__,c,m);
89     if(!nextLine(from) || !readVal(tempVals[i].scalar06)) readError(__LINE__,c,m);
90     if(!nextLine(from) || !readVal(tempVals[i].scalar07)) readError(__LINE__,c,m);
91     if(!nextLine(from) || !readVal(tempVals[i].scalar08)) readError(__LINE__,c,m);
92     if(!nextLine(from) || !readVal(tempVals[i].scalar09)) readError(__LINE__,c,m);
93     if(!nextLine(from) || !readVal(tempVals[i].scalar10)) readError(__LINE__,c,m);
94     if(!nextLine(from) || !readVal(tempVals[i].scalar11)) readError(__LINE__,c,m);
95     if(!nextLine(from) || !readVal(tempVals[i].scalar12)) readError(__LINE__,c,m);
96     if(!nextLine(from) || !readVal(tempVals[i].scalar13)) readError(__LINE__,c,m);
97     if(!nextLine(from) || !readVal(tempVals[i].scalar14)) readError(__LINE__,c,m);
98     if(!nextLine(from) || !readVal(tempVals[i].scalar15)) readError(__LINE__,c,m);
99     if(!nextLine(from) || !readVal(tempVals[i].scalar16)) readError(__LINE__,c,m);
100
101 /* example format
102  * for(int i=0;i<NUM_DB_ROWS;i++){

```

```

103 * if(!nextLine(from) || !readVal(tempVals[i].blah)) readError(l,c,m);
104 *
105 *     ....
106 *
107 * }
108 */
109
110 from.close();
111 return true;
112 #undef __METHOD__
113 }
```

### 3.2.2.11 bool rhicScalarsSender::readData (const char \**fileName*) [virtual]

Definition at line 115 of file rhicScalarsSender.cc.

```

115 {
116 #define __METHOD__ "readData(fileName)"
117
118 ifstream from(fileName);
119 if(!from) return sendMess("Cannot open file=",fileName,dbMErr,__LINE__,__CLASS__,__METHOD__);
120
121 return readData(from); // user implemented file read
122 #undef __METHOD__
123 }
```

### 3.2.2.12 void rhicScalarsSender::readError (int *l*, char \**c*, char \**m*) [inline]

Definition at line 82 of file rhicScalarsSender.hh.

```

82 {
83     mreadStatus=sendMess(" *** Missing Data at ",mline,dbMErr,l,c,m);
84 }
```

### 3.2.2.13 bool rhicScalarsSender::readVal (long long & *value*)

Definition at line 278 of file rhicScalarsSender.cc.

```

278 {
279
280     if(!readAny())return false;
281     char* store[256];
282     value=strtoll(ptr2,store,10);
283     if(strlen(*store)>0) return false; // value is not a number
284
285     return true;
286 };
```

### 3.2.2.14 bool rhicScalarsSender::readVal (long & *value*)

Definition at line 267 of file rhicScalarsSender.cc.

```

267
268
269     if(!readAny())return false;
270
271     char* store[256];
272     value=strtol(ptr2,store,10);
273     if(strlen(*store)>0) return false; // value is not a number
274
275     return true;
276 };

```

**3.2.2.15 bool rhicScalarsSender::readVal (unsigned int & *value*)**

Definition at line 256 of file rhicScalarsSender.cc.

```

256
257
258     if(!readAny()) return false;
259
260     char* store[256];
261     value=(unsigned int)strtol(ptr2,store,10);
262     if(strlen(*store)>0) return false; // value is not a number
263
264     return true;
265 };

```

**3.2.2.16 bool rhicScalarsSender::readVal (int & *value*)**

Definition at line 246 of file rhicScalarsSender.cc.

```

246
247
248     if(!readAny()) return false;
249
250     char* store[256];
251     value=(int)strtol(ptr2,store,10);
252     if(strlen(*store)>0) return false; // value is not a number
253
254     return true;
255 };

```

**3.2.2.17 bool rhicScalarsSender::readVal (short & *value*)**

Definition at line 235 of file rhicScalarsSender.cc.

```

235
236
237     if(!readAny()) return false;
238
239     char* store[256];
240     value=(short)strtol(ptr2,store,10);
241     if(strlen(*store)>0) return false; // value is not a number
242
243     return true;
244 };

```

### 3.2.2.18 bool rhicScalarsSender::readVal (double & *value*)

Definition at line 224 of file rhicScalarsSender.cc.

```
224                                         {
225
226     if(!readAny())return false;
227
228     char* store[256];
229     value=strtod(ptr2,store);
230     if(strlen(*store)>0) return false; // value is not a number
231
232     return true;
233 };
```

### 3.2.2.19 bool rhicScalarsSender::readVal (float & *value*)

Definition at line 213 of file rhicScalarsSender.cc.

```
213                                         {
214
215     if(!readAny()) return false;
216
217     char* store[256];
218     value=(float)strtod(ptr2,store);
219     if(strlen(*store)>0) return false; // value is not a number
220
221     return true;
222 };
```

### 3.2.2.20 bool rhicScalarsSender::readVal (char \*& *value*)

Definition at line 206 of file rhicScalarsSender.cc.

```
206                                         {
207
208     if(!readAny()) return false;
209     strcpy(value,ptr2);
210
211 }
```

### 3.2.2.21 bool rhicScalarsSender::updateDb (const char \**fileName*) [virtual]

Definition at line 126 of file rhicScalarsSender.cc.

```
126                                         {
127 #define __METHOD__ "updateDb(filename)"
128
129     if(!readData(fileName)) return sendMess(" Read data failed",dbMErr,__LINE__,__CLASS__,__METHOD__);
130
131     int* elements;
132     rhicScalars* vals;
133     int numRows = 0;
134
135     if(writeRequired()){
136 }
```

```

137     numRows=NUM_DB_ROWS;
138     elements=elementList;
139     vals = tempVals;
140
141 } else {
142
143     for(int i=0; i<NUM_DB_ROWS; i++){
144         if(hasChanged(i)){
145             updateElements[numRows]=elementList[i];
146             updateVals[numRows] = tempVals[i];
147             previousVals[i]=tempVals[i];
148             numRows++;
149         }
150     }
151
152     elements = updateElements;
153     vals      = updateVals;
154 }
155
156 if(numRows==0) return sendMess(" No update required for",mibName,dbMDebug,__LINE__,__CLASS__,__METHOD__);
157
158 //char mess[256];
159 ostringstream sn;
160 sn<<"Will Update "<<numRows<< " of " <<NUM_DB_ROWS<<" rows "<<ends;
161 sendMess((sn.str()).c_str(),dbMDebug,__LINE__,__CLASS__,__METHOD__);
162
163 StDbTable* dbTable=node->findTable("rhicScalars");
164 dbTable->SetTable((char*)vals, numRows, elements);
165 mgr->setStoreTime(writeTime);
166
167 if(!mgr->storeDbTable(dbTable)) {
168     addBackLog(writeTime);
169     return sendMess("Store failed ",dbMErr,__LINE__,__CLASS__,__METHOD__);
170 }
171
172 if(numRows==NUM_DB_ROWS)lastFullWrite=writeTime;
173
174 return true;
175 #undef __METHOD__
176 }
```

### 3.2.3 Member Data Documentation

#### 3.2.3.1 int rhicScalarsSender::countLimit [protected]

dito

Definition at line 36 of file rhicScalarsSender.hh.

#### 3.2.3.2 int rhicScalarsSender::elementList[NUM\_DB\_ROWS] [protected]

Definition at line 24 of file rhicScalarsSender.hh.

#### 3.2.3.3 unsigned int rhicScalarsSender::lastTime [protected]

Definition at line 37 of file rhicScalarsSender.hh.

**3.2.3.4 char [rhicScalarsSender::mline\[256\]](#) [protected]**

Definition at line 29 of file rhicScalarsSender.hh.

**3.2.3.5 bool [rhicScalarsSender::mreadStatus](#) [protected]**

Definition at line 28 of file rhicScalarsSender.hh.

**3.2.3.6 rhicScalars [rhicScalarsSender::previousVals\[NUM\\_DB\\_ROWS\]](#) [protected]**

Definition at line 22 of file rhicScalarsSender.hh.

**3.2.3.7 char\* [rhicScalarsSender::ptr1](#) [protected]**

Definition at line 31 of file rhicScalarsSender.hh.

**3.2.3.8 char \* [rhicScalarsSender::ptr2](#) [protected]**

Definition at line 31 of file rhicScalarsSender.hh.

**3.2.3.9 rhicScalars [rhicScalarsSender::tempVals\[NUM\\_DB\\_ROWS\]](#) [protected]**

Definition at line 23 of file rhicScalarsSender.hh.

**3.2.3.10 char [rhicScalarsSender::tmpline\[256\]](#) [protected]**

Definition at line 30 of file rhicScalarsSender.hh.

**3.2.3.11 int [rhicScalarsSender::updateElements\[NUM\\_DB\\_ROWS\]](#) [protected]**

Definition at line 26 of file rhicScalarsSender.hh.

**3.2.3.12 rhicScalars [rhicScalarsSender::updateVals\[NUM\\_DB\\_ROWS\]](#) [protected]**

Definition at line 25 of file rhicScalarsSender.hh.

The documentation for this class was generated from the following files:

- [rhicScalarsSender.hh](#)
- [rhicScalarsSender.cc](#)
- [rhicScalarsSender\\_i.cc](#)

### 3.3 starMagnetSender Class Reference

```
#include <starMagnetSender.hh>
```

#### Public Member Functions

- `starMagnetSender` (const char \*localDir)
- virtual ~`starMagnetSender` ()
- virtual void `initTable` ()
- virtual void `initTags` ()
- virtual void `initDataBase` ()
- virtual bool `loadUserControls` (const char \*name, const char \*value)
- virtual void `initQuery` ()
- virtual bool `queryData` ()
- virtual bool `readData` (const char \*fileName)
- virtual bool `updateDb` (const char \*fileName)
- virtual bool `readData` (ifstream &from)
- virtual bool `hasChanged` (int rowNumber)
- char \* `readAny` ()
- bool `readVal` (char \*&value)
- bool `readVal` (float &value)
- bool `readVal` (double &value)
- bool `readVal` (short &value)
- bool `readVal` (int &value)
- bool `readVal` (long &value)
- bool `readVal` (long long &value)
- bool `nextLine` (ifstream &from)
- void `readError` (int l, char \*c, char \*m)

#### Protected Attributes

- starMagnet `previousVals` [NUM\_DB\_ROWS]
- starMagnet `tempVals` [NUM\_DB\_ROWS]
- int `elementList` [NUM\_DB\_ROWS]
- starMagnet `updateVals` [NUM\_DB\_ROWS]
- int `updateElements` [NUM\_DB\_ROWS]
- bool `mreadStatus`
- char `mline` [256]
- char `tmpline` [256]
- char \* `ptr1`
- char \* `ptr2`
- float `cdriftLimit`  
*dito*
- float `pttdriftLimit`
- float `trimdriftLimit`
- int `linkstatus`

### 3.3.1 Constructor & Destructor Documentation

#### 3.3.1.1 starMagnetSender::starMagnetSender (const char \* *localDir*)

Definition at line 19 of file starMagnetSender.cc.

```

19
20
21     initTags();
22     if(localDir) cd(localDir); // note this ignores the sub dir tag
23     init("starMagnet"); // setup the file I/O
24     initDataBase();      // database connections
25     initTable();         // table definitions
26
27 }
```

#### 3.3.1.2 virtual starMagnetSender::~starMagnetSender () [inline, virtual]

Definition at line 45 of file starMagnetSender.hh.

```
45 {};
```

### 3.3.2 Member Function Documentation

#### 3.3.2.1 bool starMagnetSender::hasChanged (int *rowNumber*) [virtual]

Definition at line 146 of file starMagnetSender\_i.cc.

```

146
147
148     if(!mreadStatus) return false; // don't update on read failure
149
150
151 starMagnet* pre=&previousVals[rowNumber];
152 starMagnet* cur=&tempVals[rowNumber];
153
154 if(cur->linkStatus>=linkstatus) return false; // don't update on link failed
155
156 if(fabs(pre->mainMagnetCurrent-cur->mainMagnetCurrent)>=cdriftLimit) return true;
157 if(fabs(pre->pttWestCurrent-cur->pttWestCurrent)>=pttdriftLimit) return true;
158 if(fabs(pre->pttEastCurrent-cur->pttEastCurrent)>=pttdriftLimit) return true;
159 if(fabs(pre->trimWestCurrent-cur->trimWestCurrent)>=trimdriftLimit) return true;
160 if(fabs(pre->trimEastCurrent-cur->trimEastCurrent)>=trimdriftLimit) return true;
161
162 if((pre->pttWestStatus!=cur->pttWestStatus))return true;
163 if((pre->pttEastStatus!=cur->pttEastStatus))return true;
164 if((pre->trimWestStatus!=cur->trimWestStatus))return true;
165 if((pre->trimEastStatus!=cur->trimEastStatus))return true;
166
167
168 /* example ... note -> change to any element requires db-update
169 * and thus returns true immediately
170 *
171 *if(fabs(pre->ch0Voltage-cur->ch0Voltage)>=driftLimit) return true;
172 *if(fabs(pre->ch1Voltage-cur->ch1Voltage)>=driftLimit) return true;
173 *
174 */
175
176 }
```

```
177     return false;
178 }
```

### 3.3.2.2 void starMagnetSender::initDataBase () [virtual]

Definition at line 75 of file starMagnetSender.cc.

```
75 {
76 #define __METHOD__ "initDataBase()"
77
78 /* More than an example... swap user & dbTrg as per subsystem*/
79 mgr->setUser("stardb","");
80 StDbType dbT = dbConditions;
81 StDbDomain dbD = dbRhic;
82
83 if( !( node = mgr->initConfig(dbT,dbD)) )
84     sendMess( "Connect Failed ",mgr->printDbName(dbT,dbD),dbMFatal,__LINE__,__CLASS__,__METHOD__);
85
86 #undef __METHOD__
87 }
```

### 3.3.2.3 void starMagnetSender::initQuery () [virtual]

Definition at line 59 of file starMagnetSender\_i.cc.

```
59 {
60 #define __METHOD__ "initQuery()"
61
62     ofstream to(queryFile);
63
64     if(!to.is_open()){
65         sendMess( "Open Failed ",queryFile,dbMFatal,__LINE__,__CLASS__,__METHOD__);
66         return;
67     }
68
69 /* example
70 *      for(int i=0;i<16;i++){
71 *          to<<"TRGhv:SUB_RD_V_1:"<<i<<.E"<<endl;
72 *          to<<"TRGhv:SUB_RD_V_1:"<<i<<.F"<<endl;
73 *          ....
74 *
75 *      }
76 */
77
78     to << "cdev_mainMagnet2" << endl;
79     to << "cdev_pttWest2" << endl;
80     to << "cdev_pttEast2" << endl;
81     to << "cdev_trimWest2" << endl;
82     to << "cdev_trimEast2" << endl;
83     to << "cdev_mainMagnetm" << endl;
84     to << "cdev_pttWestm" << endl;
85     to << "cdev_pttEastm" << endl;
86     to << "cdev_trimWestm" << endl;
87     to << "cdev_trimEastm" << endl;
88     to << "testword0" << endl;
89     to << "cdev_mainMagnetB" << endl;
90
91     to.close();
92
93 #undef __METHOD__
94 }
```

### 3.3.2.4 void starMagnetSender::initTable () [virtual]

Definition at line 30 of file starMagnetSender.cc.

```

30
31 #define __METHOD__ "initTable()"
32
33     StDbTable* table=0;
34     if(!(table=node->addDbTable("starMagnet")))
35         sendMess("Could not find table=starMagnet",dbMFatal,__LINE__,__CLASS__,__METHOD__);
36
37     memset(tempVals,0,NUM_DB_ROWS*sizeof(starMagnet));
38     memset(previousVals,0,NUM_DB_ROWS*sizeof(starMagnet));
39
40     int nrows;
41     int* elist = table->getElementID(nrows);
42     if(nrows!=NUM_DB_ROWS){
43         //char mess;
44         ostringstream ms;
45         ms<<"Db rows("<<nrows<<") != compiled("<<NUM_DB_ROWS<< ")"<<ends;
46         sendMess((ms.str()).c_str(),dbMFatal,__LINE__,__CLASS__,__METHOD__);
47     }
48     memcpy(elementList,elist,NUM_DB_ROWS*sizeof(int));
49
50     unsigned int timestamp=time(NULL);
51     mgr->setRequestTime(timestamp);
52     if(mgr->fetchDbTable(table)){
53         starMagnet* thv = (starMagnet*)table->GetTable();
54         memcpy(previousVals,thv,nrows*sizeof(starMagnet));
55     }
56
57 #undef __METHOD__
58 }
```

### 3.3.2.5 void starMagnetSender::initTags () [virtual]

Definition at line 66 of file starMagnetSender.cc.

```

66
67     /* more than an example -> swap "trg" to your subsys & add to email list*/
68     setEmailTo("porter@bnl.gov");
69     setDomainName("rhic");
70
71 }
```

### 3.3.2.6 bool starMagnetSender::loadUserControls (const char \* name, const char \* value) [virtual]

Definition at line 20 of file starMagnetSender\_i.cc.

```

20
21 #define __METHOD__ "loadUserControls(name,value)"
22
23 /* more than an example ... swap driftLimit to yours
24 * and duplicate this structure for each selection criteria
25     if strstr(name,"driftLimit")){
26         driftLimit=atof(value);
27         sendMess("driftLimit set=",value,dbMDebug,__LINE__,__CLASS__,__METHOD__);
28         return true;
```

```

29     }
30   */
31   if(strstr(name,"cdriftLimit")){
32     cdriftLimit=atof(value);
33     sendMess("cdriftLimit set=",value,dbMDebug,__LINE__,__CLASS__,__METHOD__);
34     return true;
35   }
36   if(strstr(name,"pttdriftLimit")){
37     pttdriftLimit=atof(value);
38     sendMess("pttdriftLimit set=",value,dbMDebug,__LINE__,__CLASS__,__METHOD__);
39     return true;
40   }
41   if(strstr(name,"trimdriftLimit")){
42     trimdriftLimit=atof(value);
43     sendMess("trimdriftLimit set=",value,dbMDebug,__LINE__,__CLASS__,__METHOD__);
44     return true;
45   }
46   if(strstr(name,"linkstatus")){
47     linkstatus=atoi(value);
48     sendMess("linkstatus set=",value,dbMDebug,__LINE__,__CLASS__,__METHOD__);
49     return true;
50   }
51
52
53 return false;
54 #undef __METHOD__
55 }
```

### 3.3.2.7 bool starMagnetSender::nextLine (ifstream & from) [inline]

Definition at line 78 of file starMagnetSender.hh.

```

78
79   if(!from.getline(mline,255))return false;
80   return true;
81 }
```

### 3.3.2.8 bool starMagnetSender::queryData () [virtual]

Definition at line 91 of file starMagnetSender.cc.

```

91
92 #define __METHOD__ "queryData()"
93
94 /*
95  * MORE THAN AN EXAMPLE....
96  * IF Standard SC-Query via "caGet" then,
97  * no need to change this method AT ALL
98  *
99 */
100
101 writeTime = (unsigned int)time(NULL);           //for database write time
102
103 //char systemCmd[1024];
104 ostringstream scmd;
105 scmd<<"caGet "<<queryFile<<" "<<dataFile<<ends;
106
107 if(system((scmd.str()).c_str()))
108   return sendMess(" caGet system call returned error",dbMErr,__LINE__,__CLASS__,__METHOD__);
109
110 return true;
```

```
111 #undef __METHOD__
112 };
```

### 3.3.2.9 char \* starMagnetSender::readAny ()

Definition at line 197 of file starMagnetSender.cc.

```
197
198
199     strcpy(tmpLine,mLine);
200     ptr1=tmpLine;
201     ptr2=strtok(ptr1," ");
202     if(!ptr2) return ptr2;
203     ptr2=strtok(NULL," ");
204     return ptr2;
205 }
```

### 3.3.2.10 bool starMagnetSender::readData (ifstream & from) [virtual]

Definition at line 99 of file starMagnetSender\_i.cc.

```
99
100 #define __METHOD__ "readData(ifstream)"
101
102 mreadStatus=true;
103 memset(tempVals,0,NUM_DB_ROWS*sizeof(starMagnet));
104
105 int i=0;
106 if(!nextLine(from)|| !readVal(tempVals[i].mainMagnetCurrent)) readError(__LINE__,__CLASS__,__METHOD__);
107 if(!nextLine(from)|| !readVal(tempVals[i].pttWestCurrent)) readError(__LINE__,__CLASS__,__METHOD__);
108 if(!nextLine(from)|| !readVal(tempVals[i].pttEastCurrent)) readError(__LINE__,__CLASS__,__METHOD__);
109 if(!nextLine(from)|| !readVal(tempVals[i].trimWestCurrent)) readError(__LINE__,__CLASS__,__METHOD__);
110 if(!nextLine(from)|| !readVal(tempVals[i].trimEastCurrent)) readError(__LINE__,__CLASS__,__METHOD__);
111
112 if(!nextLine(from)|| !readVal(tempVals[i].mainMagnetStatus)) readError(__LINE__,__CLASS__,__METHOD__);
113 if(!nextLine(from)|| !readVal(tempVals[i].pttWestStatus)) readError(__LINE__,__CLASS__,__METHOD__);
114 if(!nextLine(from)|| !readVal(tempVals[i].pttEastStatus)) readError(__LINE__,__CLASS__,__METHOD__);
115 if(!nextLine(from)|| !readVal(tempVals[i].trimWestStatus)) readError(__LINE__,__CLASS__,__METHOD__);
116 if(!nextLine(from)|| !readVal(tempVals[i].trimEastStatus)) readError(__LINE__,__CLASS__,__METHOD__);
117
118 if(!nextLine(from)|| !readVal(tempVals[i].linkStatus)) readError(__LINE__,__CLASS__,__METHOD__);
119 //MPD Changes
120     char* tmpval0;
121     if(!nextLine(from)){
122             tmpval0=tempVals[i].readoutMode;
123             if(!readVal(tmpval0))readError(__LINE__,__CLASS__,__METHOD__);
124         }
125 //if(!nextLine(from)|| !readVal(tempVals[i].readoutMode)) readError(__LINE__,__CLASS__,__METHOD__);
126
127
128 /* example format
129 *   for(int i=0;i<NUM_DB_ROWS;i++){
130 *     if(!nextLine(from) || !readVal(tempVals[i].blah)) readError(__LINE__,__CLASS__,__METHOD__);
131 *     if(!nextLine(from) || !readVal(tempVals[i].ch0Voltage)) readError(__LINE__,__CLASS__,__METHOD__);
132 *     if(!nextLine(from) || !readVal(tempVals[i].ch1Voltage)) readError(__LINE__,__CLASS__,__METHOD__);
133 *
134 *     ....
135 *
136 *   }
137 */
```

```

139     from.close();
140     return true;
141 #undef __METHOD__
142 }
```

### 3.3.2.11 bool starMagnetSender::readData (const char \**fileName*) [virtual]

Definition at line 116 of file starMagnetSender.cc.

```

116
117 #define __METHOD__ "readData(fileName)"
118
119     ifstream from(fileName);
120     if(!from) return sendMess("Cannot open file=",fileName,dbMErr,__LINE__,__CLASS__,__METHOD__);
121
122     return readData(from); // user implemented file read
123 #undef __METHOD__
124 }
```

### 3.3.2.12 void starMagnetSender::readError (int *l*, char \**c*, char \**m*) [inline]

Definition at line 83 of file starMagnetSender.hh.

```

83
84     mreadStatus=sendMess(" *** Missing Data at ",mline,dbMErr,l,c,m);
85 }
```

### 3.3.2.13 bool starMagnetSender::readVal (long long &*value*)

Definition at line 269 of file starMagnetSender.cc.

```

269
270
271     if(!readAny())return false;
272     char* store[256];
273     value=strtoll(ptr2,store,10);
274     if(strlen(*store)>0) return false; // value is not a number
275
276     return true;
277 };
```

### 3.3.2.14 bool starMagnetSender::readVal (long &*value*)

Definition at line 258 of file starMagnetSender.cc.

```

258
259
260     if(!readAny())return false;
261
262     char* store[256];
263     value=strtol(ptr2,store,10);
264     if(strlen(*store)>0) return false; // value is not a number
265
266     return true;
267 };
```

### 3.3.2.15 bool starMagnetSender::readVal (int & value)

Definition at line 247 of file starMagnetSender.cc.

```
247 {  
248  
249     if(!readAny()) return false;  
250  
251     char* store[256];  
252     value=(int)strtol(ptr2,store,10);  
253     if(strlen(*store)>0) return false; // value is not a number  
254  
255     return true;  
256 };
```

### 3.3.2.16 bool starMagnetSender::readVal (short & value)

Definition at line 236 of file starMagnetSender.cc.

```
236 {  
237  
238     if(!readAny()) return false;  
239  
240     char* store[256];  
241     value=(short)strtol(ptr2,store,10);  
242     if(strlen(*store)>0) return false; // value is not a number  
243  
244     return true;  
245 };
```

### 3.3.2.17 bool starMagnetSender::readVal (double & value)

Definition at line 225 of file starMagnetSender.cc.

```
225 {  
226  
227     if(!readAny())return false;  
228  
229     char* store[256];  
230     value=strtod(ptr2,store);  
231     if(strlen(*store)>0) return false; // value is not a number  
232  
233     return true;  
234 };
```

### 3.3.2.18 bool starMagnetSender::readVal (float & value)

Definition at line 214 of file starMagnetSender.cc.

```
214 {  
215  
216     if(!readAny()) return false;  
217  
218     char* store[256];  
219     value=(float)strtod(ptr2,store);  
220     if(strlen(*store)>0) return false; // value is not a number  
221  
222     return true;  
223 };
```

**3.3.2.19 bool starMagnetSender::readVal (char \*& *value*)**

Definition at line 207 of file starMagnetSender.cc.

```
207
208
209     if(!readAny()) return false;
210     strcpy(value,ptr2);
211     return true;
212 }
```

**3.3.2.20 bool starMagnetSender::updateDb (const char \**fileName*) [virtual]**

Definition at line 127 of file starMagnetSender.cc.

```
127
128 #define __METHOD__ "updateDb(filename)"
129
130     if(!readData(fileName)) return sendMess(" Read data failed",dbMErr,__LINE__,__CLASS__,__METHOD__);
131
132     int* elements;
133     starMagnet* vals;
134     int numRows = 0;
135
136     if(writeRequired()){
137
138         numRows=NUM_DB_ROWS;
139         elements=elementList;
140         vals = tempVals;
141
142     } else {
143
144         for(int i=0; i<NUM_DB_ROWS; i++){
145             if(hasChanged(i)){
146                 updateElements[numRows]=elementList[i];
147                 updateVals[numRows] = tempVals[i];
148                 previousVals[i]=tempVals[i];
149                 numRows++;
150             }
151         }
152
153         elements = updateElements;
154         vals      = updateVals;
155     }
156
157     if(numRows==0) return sendMess(" No update required for",mbaseName,dbMDebug,__LINE__,__CLASS__,__METHOD__);
158
159     //char mess[256];
160     ostringstream sn;
161     sn<<"Will Update "<<numRows<<" of "<<NUM_DB_ROWS<<" rows "<<ends;
162     sendMess((sn.str()).c_str(),dbMDebug,__LINE__,__CLASS__,__METHOD__);
163
164     StDbTable* dbTable=node->findTable("starMagnet");
165     dbTable->SetTable((char*)vals, numRows, elements);
166     mgr->setStoreTime(writeTime);
167
168     if(!mgr->storeDbTable(dbTable)) {
169         addBackLog(writeTime);
170         return sendMess("Store failed ",dbMErr,__LINE__,__CLASS__,__METHOD__);
171     }
172
173     if(numRows==NUM_DB_ROWS)lastFullWrite=writeTime;
174 }
```

```
175     return true;
176 #undef __METHOD__
177 }
```

### 3.3.3 Member Data Documentation

#### 3.3.3.1 float starMagnetSender::cdriftLimit [protected]

dito

Definition at line 35 of file starMagnetSender.hh.

#### 3.3.3.2 int starMagnetSender::elementList[NUM\_DB\_ROWS] [protected]

Definition at line 24 of file starMagnetSender.hh.

#### 3.3.3.3 int starMagnetSender::linkstatus [protected]

Definition at line 38 of file starMagnetSender.hh.

#### 3.3.3.4 char starMagnetSender::mline[256] [protected]

Definition at line 29 of file starMagnetSender.hh.

#### 3.3.3.5 bool starMagnetSender::mreadStatus [protected]

Definition at line 28 of file starMagnetSender.hh.

#### 3.3.3.6 starMagnet starMagnetSender::previousVals[NUM\_DB\_ROWS] [protected]

Definition at line 22 of file starMagnetSender.hh.

#### 3.3.3.7 char\* starMagnetSender::ptr1 [protected]

Definition at line 31 of file starMagnetSender.hh.

#### 3.3.3.8 char \* starMagnetSender::ptr2 [protected]

Definition at line 31 of file starMagnetSender.hh.

#### 3.3.3.9 float starMagnetSender::pttdriftLimit [protected]

Definition at line 36 of file starMagnetSender.hh.

#### 3.3.3.10 starMagnet starMagnetSender::tempVals[NUM\_DB\_ROWS] [protected]

Definition at line 23 of file starMagnetSender.hh.

**3.3.3.11 char [starMagnetSender::tmpLine\[256\]](#) [protected]**

Definition at line 30 of file starMagnetSender.hh.

**3.3.3.12 float [starMagnetSender::trimDriftLimit](#) [protected]**

Definition at line 37 of file starMagnetSender.hh.

**3.3.3.13 int [starMagnetSender::updateElements\[NUM\\_DB\\_ROWS\]](#) [protected]**

Definition at line 26 of file starMagnetSender.hh.

**3.3.3.14 starMagnet [starMagnetSender::updateVals\[NUM\\_DB\\_ROWS\]](#) [protected]**

Definition at line 25 of file starMagnetSender.hh.

The documentation for this class was generated from the following files:

- [starMagnetSender.hh](#)
- [starMagnetSender.cc](#)
- [starMagnetSender\\_i.cc](#)



# Chapter 4

## Doxxygen\_MPDAEMON\_StarOnlineDb File Documentation

### 4.1 rhicBeamDaemon.cc File Reference

```
#include "rhicBeamSender.hh"
#include <unistd.h>
```

#### Functions

- void [runSender](#) (const char \**ldir*)

#### 4.1.1 Function Documentation

##### 4.1.1.1 void [runSender](#) (const char \* *ldir*)

Definition at line 14 of file rhicBeamDaemon.cc.

```
14
15
16 CndDbSender* sender = new rhicBeamSender(ldir);
17
18 sender->initQuery();
19 for(;;) { //ever...
20     if(sender->hasBackLog())sender->cleanBackLog();
21     if(sender->queryData())sender->updateDb();
22     sleep(sender->sleepTime());
23 }
24
25 };
```

## 4.2 rhicBeamSender.cc File Reference

```
#include <stdlib.h>
#include <unistd.h>
#include "rhicBeamSender.hh"
#include "StDbTable.h"
#include "rhicBeamSender_i.cc"
```

### Defines

- #define **\_CLASS\_** "rhicBeamSender"
- #define **\_METHOD\_** "initTable()
- #define **\_METHOD\_** "initDataBase()
- #define **\_METHOD\_** "queryData()
- #define **\_METHOD\_** "readData(fileName)"
- #define **\_METHOD\_** "updateDb(filename)"

### 4.2.1 Define Documentation

#### 4.2.1.1 #define **\_CLASS\_** "rhicBeamSender"

Definition at line 16 of file rhicBeamSender.cc.

#### 4.2.1.2 #define **\_METHOD\_** "updateDb(filename)"

#### 4.2.1.3 #define **\_METHOD\_** "readData(fileName)"

#### 4.2.1.4 #define **\_METHOD\_** "queryData()"

#### 4.2.1.5 #define **\_METHOD\_** "initDataBase()"

#### 4.2.1.6 #define **\_METHOD\_** "initTable()"

## 4.3 rhicBeamSender.hh File Reference

```
#include "CndDbSender.hh"
#include "rhicBeam.h"
```

### Classes

- class [rhicBeamSender](#)

### Defines

- #define [NUM\\_DB\\_ROWS](#) 1

#### 4.3.1 Define Documentation

##### 4.3.1.1 #define NUM\_DB\_ROWS 1

Definition at line 16 of file rhicBeamSender.hh.

## 4.4 rhicBeamSender\_i.cc File Reference

### Defines

- #define METHOD\_ "loadUserControls(name,value)"
- #define METHOD\_ "initQuery()"
- #define METHOD\_ "readData(ifstream)"

#### 4.4.1 Define Documentation

4.4.1.1 #define METHOD\_ "readData(ifstream)"

4.4.1.2 #define METHOD\_ "initQuery()"

4.4.1.3 #define METHOD\_ "loadUserControls(name,value)"

## 4.5 rhicScalarsDaemon.cc File Reference

```
#include "rhicScalarsSender.hh"
#include <unistd.h>
```

### Functions

- void **runSender** (const char \*ldir)

#### 4.5.1 Function Documentation

##### 4.5.1.1 void runSender (const char \* *ldir*)

Definition at line 14 of file rhicScalarsDaemon.cc.

```
14
15
16     CndDbSender* sender = new rhicScalarsSender(ldir);
17
18     sender->initQuery();
19     for(;;) { /*ever...
20         if(sender->hasBackLog())sender->cleanBackLog();
21         if(sender->queryData())sender->updateDb();
22         sleep(sender->sleepTime());
23     }
24
25 };
```

## 4.6 rhicScalarsSender.cc File Reference

```
#include <stdlib.h>
#include <unistd.h>
#include "rhicScalarsSender.hh"
#include "StDbTable.h"
#include "rhicScalarsSender_i.cc"
```

### Defines

- #define **\_CLASS\_** "rhicScalarsSender"
- #define **\_METHOD\_** "initTable()"
- #define **\_METHOD\_** "initDataBase()"
- #define **\_METHOD\_** "queryData()"
- #define **\_METHOD\_** "readData(fileName)"
- #define **\_METHOD\_** "updateDb(filename)"

#### 4.6.1 Define Documentation

##### 4.6.1.1 #define **\_CLASS\_** "rhicScalarsSender"

Definition at line 16 of file rhicScalarsSender.cc.

##### 4.6.1.2 #define **\_METHOD\_** "updateDb(filename)"

##### 4.6.1.3 #define **\_METHOD\_** "readData(fileName)"

##### 4.6.1.4 #define **\_METHOD\_** "queryData()"

##### 4.6.1.5 #define **\_METHOD\_** "initDataBase()"

##### 4.6.1.6 #define **\_METHOD\_** "initTable()"

## 4.7 rhicScalarsSender.hh File Reference

```
#include "CndDbSender.hh"
#include "rhicScalars.h"
```

### Classes

- class [rhicScalarsSender](#)

### Defines

- #define [NUM\\_DB\\_ROWS](#) 1

#### 4.7.1 Define Documentation

##### 4.7.1.1 #define NUM\_DB\_ROWS 1

Definition at line 16 of file rhicScalarsSender.hh.

## 4.8 rhicScalarsSender\_i.cc File Reference

### Defines

- #define METHOD\_ "loadUserControls(name,value)"
- #define METHOD\_ "initQuery()"
- #define METHOD\_ "readData(ifstream)"

#### 4.8.1 Define Documentation

4.8.1.1 #define METHOD\_ "readData(ifstream)"

4.8.1.2 #define METHOD\_ "initQuery()"

4.8.1.3 #define METHOD\_ "loadUserControls(name,value)"

## 4.9 sms\_i.cc File Reference

```
#include "starMagnetSender_i.hh"
```

### Defines

- #define **\_CLASS\_** "starMagnetSender\_i"
- #define **\_METHOD\_** "initDataBase()"
- #define **\_METHOD\_** "loadUserControls(name,value)"
- #define **\_METHOD\_** "queryData()"
- #define **\_METHOD\_** "readData(ifstream)"

### 4.9.1 Define Documentation

#### 4.9.1.1 #define **\_CLASS\_** "starMagnetSender\_i"

Definition at line 11 of file sms\_i.cc.

#### 4.9.1.2 #define **\_METHOD\_** "readData(ifstream)"

#### 4.9.1.3 #define **\_METHOD\_** "queryData()"

#### 4.9.1.4 #define **\_METHOD\_** "loadUserControls(name,value)"

#### 4.9.1.5 #define **\_METHOD\_** "initDataBase()"

## 4.10 starMagnetDaemon.cc File Reference

```
#include "starMagnetSender.hh"
#include <unistd.h>
```

### Functions

- void **runSender** (const char \*ldir)

#### 4.10.1 Function Documentation

##### 4.10.1.1 void runSender (const char \* *ldir*)

Definition at line 14 of file starMagnetDaemon.cc.

```
14
15
16     CndDbSender* sender = new starMagnetSender(ldir);
17
18     sender->initQuery();
19     for(;;) { /*ever...
20         if(sender->hasBackLog())sender->cleanBackLog();
21         if(sender->queryData())sender->updateDb();
22         sleep(sender->sleepTime());
23     }
24
25 };
```

## 4.11 starMagnetSender.cc File Reference

```
#include <stdlib.h>
#include <unistd.h>
#include <math.h>
#include "starMagnetSender.hh"
#include "StDbTable.h"
#include "starMagnetSender_i.cc"
```

### Defines

- #define \_\_CLASS\_\_ "starMagnetSender"
- #define \_\_METHOD\_\_ "initTable()"
- #define \_\_METHOD\_\_ "initDataBase()"
- #define \_\_METHOD\_\_ "queryData()"
- #define \_\_METHOD\_\_ "readData(fileName)"
- #define \_\_METHOD\_\_ "updateDb(filename)"

### 4.11.1 Define Documentation

#### 4.11.1.1 #define \_\_CLASS\_\_ "starMagnetSender"

Definition at line 17 of file starMagnetSender.cc.

#### 4.11.1.2 #define \_\_METHOD\_\_ "updateDb(filename)"

#### 4.11.1.3 #define \_\_METHOD\_\_ "readData(fileName)"

#### 4.11.1.4 #define \_\_METHOD\_\_ "queryData()"

#### 4.11.1.5 #define \_\_METHOD\_\_ "initDataBase()"

#### 4.11.1.6 #define \_\_METHOD\_\_ "initTable()"

## 4.12 starMagnetSender.hh File Reference

```
#include "CndDbSender.hh"
#include "starMagnet.h"
```

### Classes

- class [starMagnetSender](#)

### Defines

- #define [NUM\\_DB\\_ROWS](#) 1

#### 4.12.1 Define Documentation

##### 4.12.1.1 #define NUM\_DB\_ROWS 1

Definition at line 16 of file starMagnetSender.hh.

## 4.13 starMagnetSender\_i.cc File Reference

### Defines

- #define `__METHOD__` "loadUserControls(name,value)"
- #define `__METHOD__` "initQuery()"
- #define `__METHOD__` "readData(ifstream)"

#### 4.13.1 Define Documentation

4.13.1.1 #define `__METHOD__` "readData(ifstream)"

4.13.1.2 #define `__METHOD__` "initQuery()"

4.13.1.3 #define `__METHOD__` "loadUserControls(name,value)"

# Index

~rhicBeamSender  
    rhicBeamSender, 6

~rhicScalarsSender  
    rhicScalarsSender, 17

~starMagnetSender  
    starMagnetSender, 28

\_\_CLASS\_\_  
    rhicBeamSender.cc, 40  
    rhicScalarsSender.cc, 44  
    sms\_i.cc, 47  
    starMagnetSender.cc, 49

\_\_METHOD\_\_  
    rhicBeamSender.cc, 40  
    rhicBeamSender\_i.cc, 42  
    rhicScalarsSender.cc, 44  
    rhicScalarsSender\_i.cc, 46  
    sms\_i.cc, 47  
    starMagnetSender.cc, 49  
    starMagnetSender\_i.cc, 51

cdriftLimit  
    starMagnetSender, 36

countLimit  
    rhicScalarsSender, 25

elementList  
    rhicBeamSender, 14  
    rhicScalarsSender, 25  
    starMagnetSender, 36

hasChanged  
    rhicBeamSender, 6  
    rhicScalarsSender, 17  
    starMagnetSender, 28

initDataBase  
    rhicBeamSender, 7  
    rhicScalarsSender, 18  
    starMagnetSender, 29

initQuery  
    rhicBeamSender, 7  
    rhicScalarsSender, 18  
    starMagnetSender, 29

initTable  
    rhicBeamSender, 8  
    rhicScalarsSender, 19

    starMagnetSender, 29

initTags  
    rhicBeamSender, 8  
    rhicScalarsSender, 19  
    starMagnetSender, 30

lastTime  
    rhicScalarsSender, 25

linkstatus  
    starMagnetSender, 36

loadUserControls  
    rhicBeamSender, 8  
    rhicScalarsSender, 20  
    starMagnetSender, 30

mline  
    rhicBeamSender, 14  
    rhicScalarsSender, 25  
    starMagnetSender, 36

mreadStatus  
    rhicBeamSender, 14  
    rhicScalarsSender, 26  
    starMagnetSender, 36

nextLine  
    rhicBeamSender, 9  
    rhicScalarsSender, 20  
    starMagnetSender, 31

NUM\_DB\_ROWS  
    rhicBeamSender.hh, 41  
    rhicScalarsSender.hh, 45  
    starMagnetSender.hh, 50

previousVals  
    rhicBeamSender, 14  
    rhicScalarsSender, 26  
    starMagnetSender, 36

ptr1  
    rhicBeamSender, 14  
    rhicScalarsSender, 26  
    starMagnetSender, 36

ptr2  
    rhicBeamSender, 14  
    rhicScalarsSender, 26  
    starMagnetSender, 36

pttdriftLimit

starMagnetSender, 36

queryData  
  rhicBeamSender, 9  
  rhicScalarsSender, 20  
  starMagnetSender, 31

readAny  
  rhicBeamSender, 9  
  rhicScalarsSender, 21  
  starMagnetSender, 32

readData  
  rhicBeamSender, 10, 11  
  rhicScalarsSender, 21, 22  
  starMagnetSender, 32, 33

readError  
  rhicBeamSender, 11  
  rhicScalarsSender, 22  
  starMagnetSender, 33

readVal  
  rhicBeamSender, 11–13  
  rhicScalarsSender, 22–24  
  starMagnetSender, 33, 34

rhicBeamDaemon.cc, 39

rhicBeamDaemon.cc  
  runSender, 39

rhicBeamSender, 5  
  rhicBeamSender, 6

rhicBeamSender  
  ~rhicBeamSender, 6  
  elementList, 14  
  hasChanged, 6  
  initDataBase, 7  
  initQuery, 7  
  initTable, 8  
  initTags, 8  
  loadUserControls, 8  
  mline, 14  
  mreadStatus, 14  
  nextLine, 9  
  previousVals, 14  
  ptr1, 14  
  ptr2, 14  
  queryData, 9  
  readAny, 9  
  readData, 10, 11  
  readError, 11  
  readVal, 11–13  
  rhicBeamSender, 6  
  tempVals, 14  
  tmpline, 14  
  updateDb, 13  
  updateElements, 15  
  updateVals, 15

rhicBeamSender.cc, 40  
rhicBeamSender.cc  
  \_\_CLASS\_\_, 40  
  \_\_METHOD\_\_, 40

rhicBeamSender.hh, 41

rhicBeamSender.hh  
  NUM\_DB\_ROWS, 41

rhicBeamSender\_i.cc, 42

rhicBeamSender\_i.cc  
  \_\_METHOD\_\_, 42

rhicScalarsDaemon.cc, 43

rhicScalarsDaemon.cc  
  runSender, 43

rhicScalarsSender, 16  
  rhicScalarsSender, 17

rhicScalarsSender  
  ~rhicScalarsSender, 17  
  countLimit, 25  
  elementList, 25  
  hasChanged, 17  
  initDataBase, 18  
  initQuery, 18  
  initTable, 19  
  initTags, 19  
  lastTime, 25  
  loadUserControls, 20  
  mline, 25  
  mreadStatus, 26  
  nextLine, 20  
  previousVals, 26  
  ptr1, 26  
  ptr2, 26  
  queryData, 20  
  readAny, 21  
  readData, 21, 22  
  readError, 22  
  readVal, 22–24  
  rhicScalarsSender, 17  
  tempVals, 26  
  tmpline, 26  
  updateDb, 24  
  updateElements, 26  
  updateVals, 26

rhicScalarsSender.cc, 44

rhicScalarsSender.cc  
  \_\_CLASS\_\_, 44  
  \_\_METHOD\_\_, 44

rhicScalarsSender.hh, 45

rhicScalarsSender.hh  
  NUM\_DB\_ROWS, 45

rhicScalarsSender\_i.cc, 46

rhicScalarsSender\_i.cc  
  \_\_METHOD\_\_, 46

runSender

rhicBeamDaemon.cc, 39  
 rhicScalarsDaemon.cc, 43  
 starMagnetDaemon.cc, 48  
  
 sms\_i.cc, 47  
     \_\_CLASS\_\_, 47  
     \_\_METHOD\_\_, 47  
 starMagnetDaemon.cc, 48  
 starMagnetDaemon.cc  
     runSender, 48  
 starMagnetSender, 27  
     starMagnetSender, 28  
 starMagnetSender  
     ~starMagnetSender, 28  
     cdriftLimit, 36  
     elementList, 36  
     hasChanged, 28  
     initDataBase, 29  
     initQuery, 29  
     initTable, 29  
     initTags, 30  
     linkstatus, 36  
     loadUserControls, 30  
     mline, 36  
     mreadStatus, 36  
     nextLine, 31  
     previousVals, 36  
     ptr1, 36  
     ptr2, 36  
     pttdriftLimit, 36  
     queryData, 31  
     readAny, 32  
     readData, 32, 33  
     readError, 33  
     readVal, 33, 34  
     starMagnetSender, 28  
     tempVals, 36  
     tmpline, 36  
     trimdriftLimit, 37  
     updateDb, 35  
     updateElements, 37  
     updateVals, 37  
 starMagnetSender.cc, 49  
 starMagnetSender.cc  
     \_\_CLASS\_\_, 49  
     \_\_METHOD\_\_, 49  
 starMagnetSender.hh, 50  
 starMagnetSender.hh  
     NUM\_DB\_ROWS, 50  
 starMagnetSender\_i.cc, 51  
 starMagnetSender\_i.cc  
     \_\_METHOD\_\_, 51  
  
 tempVals